UNIVERSITETET I BERGEN

Det samfunnsvitenskapelige fakultet

INFO216 v2016: Project submission

Candidate 139, May 20, 2016

Project description

Purpose

It is a system used to model some situation during an Apollo 13 mission. The model consists of a *Truth Maintenance System* (loosely based on Doyle) and annotations using *rdfs:seeAlso*. The *TMS* allows *beliefs* in a *knowledge base* to be manipulated over time. The annotation system can be used to keep track of *why* some belief is held/not held, as well as arbitrary data.

Why

The Apollo 13 mission is interesting. Even though the people involved performed stellarly, mistakes were made. Some mistakes were trivial and I wanted to see if a situation could be modelled to avoid those mistakes.

Why semantic technologies

For the TMS I have used some *inference*. I.e. a user can say "*A* is in the *IN-list* of B"—and through inference we will know that *A* is a belief and *B* is a *justification*. There are several of these inferences possible in the model. When we make the model smarter, the *app* becomes smaller.

There is also an example of fetching data from *DBPedia* in the app. Merging knowledge is difficult to *do well* using non-semantic technologies.

Tools, standards, vocabularies

- I have lifted some data using python (namely the voice transcripts). I chose python because it's a great scripting language.
- Both the knowledge base and the reasoner are stored in Fuseki, each in its own graph. The default graph was not used. Sparql has been used to query the TDB via Fuseki, but not all queries are made against the TBD directly—some are made using the Jena Model API. This mix is almost random, except for a few cases where SPARQL was easier. The Fuseki db is updated whenever necessary.
- Data appear in both Turtle and N3. I chose these formats because they are quite common.
- I have used RDFS, Provenance, Simple Event Model, a self made vocabulary (JTMS) and several others vocabularies. Provenance was used to annotate; SEM to model the voice transcripts. JTMS was made to model the TMS.
- Java libraries used are Jena, MigLayout and JGrahpx. Jena was the best semantic library I could find, and MigLayout and JGraphx were used for their graphical abilities.
- The code is object oriented loosely following a *model-view-controller* pattern. I've chosen this so that I can logically separate the three.

Reading/converting/lifting

See appendix A for a flow diagram of data from various sources.

Class diagram

See appendix B for a class diagram.

Future implications

I've had difficulty learning the Jena API, hence the various ways to obtain data (*sparql*, *listStatements*). Next time I'll start a couple of other projects using Jena, and I will also join a group. I have done everything on my own, which have reduced the quality of the project.

The project code quality is low—no unit tests, highly inconsistent API usage and, at times, cryptic logic. Me being alone on the project is partly to blame. It is also due to the fact that my idea of an implementation changed over the course of the semester. Joining a group would put more pressure on developing an idea sooner (in most cases), and it would also reduce the stress of the felling of not having enough time. I will do so for my next group project.

Appendices

A Flowchart

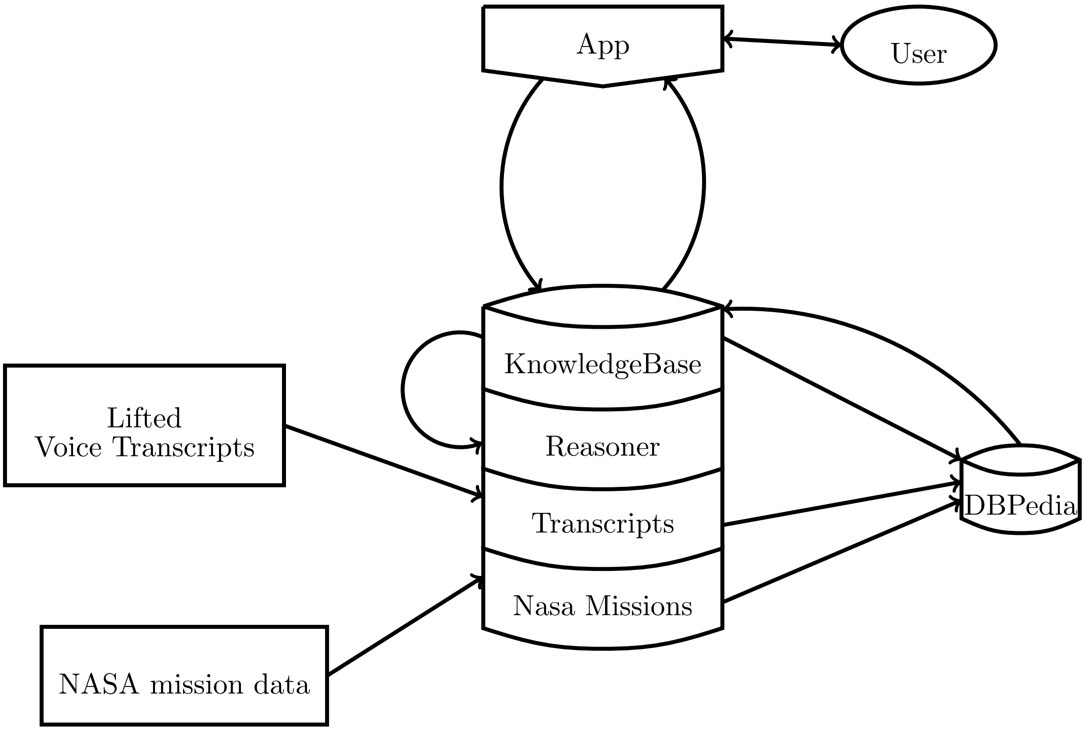
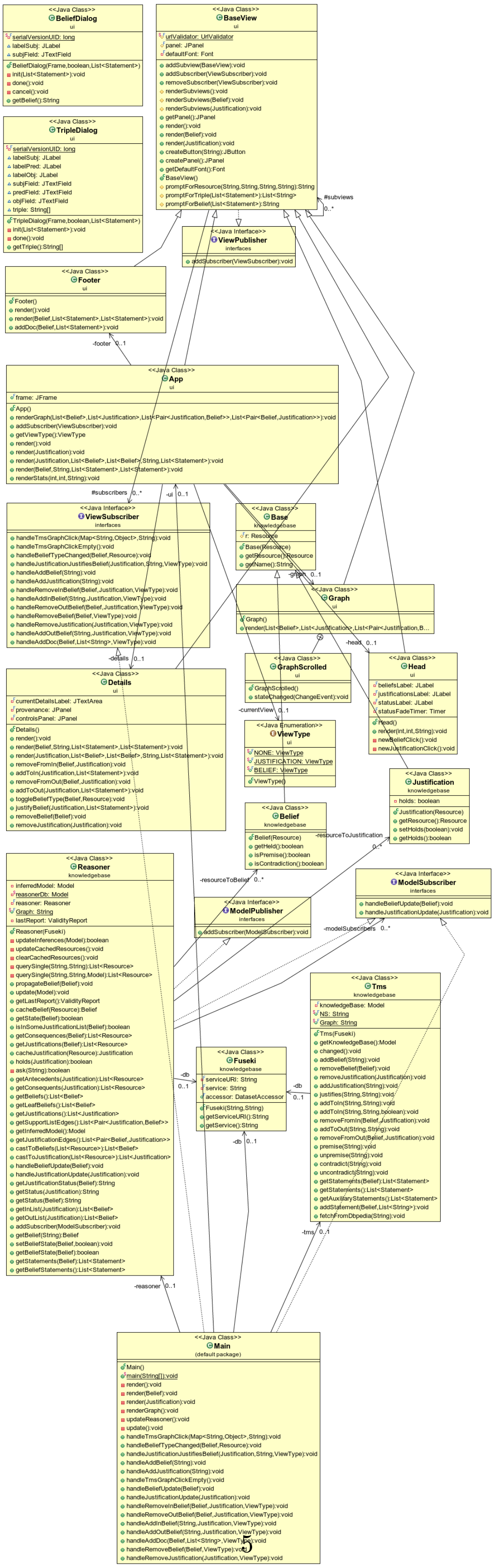


Figure 1: Flow chart

B Class diagram



<<Java Class>>
BeliefDialog
ui
serialVersionUID: long
labelSubj: JLabel
subjField: JTextField
BeliefDialog(Frame,boolean,List<Statement>)
init(List<Statement>):void
done():void
cancel():void
getBelief():String
<<Java Class>>
BaseView
ui
urlValidator: UrlValidator
panel: JPanel
defaultFont: Font
addSubview(BaseView):void
addSubscriber(ViewSubscriber):void
removeSubscriber(ViewSubscriber):void
renderSubviews():void
renderSubviews(Belief):void
renderSubviews(Justification):void
getPanel():JPanel
render():void
render(Belief):void
render(Justification):void
createButton(String):JButton
createPanel():JPanel
getDefaultFont():Font
BaseView()
promptForResource(String,String,String,String):String
promptForTriple(List<Statement>):List<String>
promptForBelief(List<Statement>):String
#subviews
0..*
<<Java Class>>
TripleDialog
ui
serialVersionUID: long
labelSubj: JLabel
labelPred: JLabel
labelObj: JLabel
subjField: JTextField
predField: JTextField
objField: JTextField
triple: String[]
TripleDialog(Frame,boolean,List<Statement>)
init(List<Statement>):void
done():void
getTriple():String[]
<<Java Interface>>
ViewPublisher
interfaces
addSubscriber(ViewSubscriber):void
<<Java Class>>
Footer
ui
Footer()
render():void
render(Belief,List<Statement>,List<Statement>):void
addDoc(Belief,List<Statement>):void
-footer
0..1
<<Java Class>>
App
ui
frame: JFrame
App()
renderGraph(List<Belief>,List<Justification>,List<Pair<Justification,Belief>>,List<Pair<Belief,Justification>>):void
addSubscriber(ViewSubscriber):void
getViewType():ViewType
render():void
render(Justification):void
render(Justification,List<Belief>,List<Belief>,String,List<Statement>):void
render(Belief,String,List<Statement>,List<Statement>):void
renderStats(int,int,String):void
#subscribers
0..*
-ui
0..1
<<Java Interface>>
ViewSubscriber
interfaces
handleTmsGraphClick(Map<String,Object>,String):void
handleTmsGraphClickEmpty():void
handleBeliefTypeChanged(Belief,Resource):void
handleJustificationJustifiesBelief(Justification,String,ViewType):void
handleAddBelief(String):void
handleAddJustification(String):void
handleRemoveInBelief(Belief,Justification,ViewType):void
handleAddInBelief(String,Justification,ViewType):void
handleRemoveOutBelief(Belief,Justification,ViewType):void
handleRemoveBelief(Belief,ViewType):void
handleRemoveJustification(Justification,ViewType):void
handleAddOutBelief(String,Justification,ViewType):void
handleAddDoc(Belief,List<String>,ViewType):void
<<Java Class>>
Base
knowledgebase
r: Resource
Base(Resource)
getResource():Resource
getName():String
-graph
0..1
<<Java Class>>
Graph
ui
Graph()
render(List<Belief>,List<Justification>,List<Pair<Justification,B...
-head
0..1
-details
0..1
<<Java Class>>
Details
ui
currentDetailsLabel: JTextArea
provenance: JPanel
controlsPanel: JPanel
Details()
render():void
render(Belief,String,List<Statement>,List<Statement>):void
render(Justification,List<Belief>,List<Belief>,String,List<Statement>):void
removeFromIn(Belief,Justification):void
addToIn(Justification,List<Statement>):void
removeFromOut(Belief,Justification):void
addToOut(Justification,List<Statement>):void
toggleBeliefType(Belief,Resource):void
justifyBelief(Justification,List<Statement>):void
removeBelief(Belief):void
removeJustification(Justification):void
<<Java Class>>
GraphScrolled
ui
GraphScrolled()
stateChanged(ChangeEvent):void
<<Java Class>>
Head
ui
beliefsLabel: JLabel
justificationsLabel: JLabel
statusLabel: JLabel
statusFadeTimer: Timer
Head()
render(int,int,String):void
newBeliefClick():void
newJustificationClick():void
-currentView
0..1
<<Java Enumeration>>
ViewType
ui
NONE: ViewType
JUSTIFICATION: ViewType
BELIEF: ViewType
ViewType()
<<Java Class>>
Justification
knowledgebase
holds: boolean
Justification(Resource)
getResource():Resource
setHolds(boolean):void
getHolds():boolean
<<Java Class>>
Belief
knowledgebase
Belief(Resource)
getHeld():boolean
isPremise():boolean
isContradiction():boolean
-resourceToJustification
0..*
-resourceToBelief
0..*
<<Java Interface>>
ModelSubscriber
interfaces
handleBeliefUpdate(Belief):void
handleJustificationUpdate(Justification):void
<<Java Interface>>
ModelPublisher
interfaces
addSubscriber(ModelSubscriber):void
-modelSubscribers
0..*
<<Java Class>>
Reasoner
knowledgebase
inferredModel: Model
reasonerDb: Model
reasoner: Reasoner
Graph: String
lastReport: ValidityReport
Reasoner(Fuseki)
updateInferences(Model):boolean
updateCachedResources():void
clearCachedResources():void
querySingle(String,String):List<Resource>
querySingle(String,String,Model):List<Resource>
propagateBelief(Belief):void
update(Model):void
getLastReport():ValidityReport
cacheBelief(Resource):Belief
getState(Belief):boolean
isInSomeJustificationList(Belief):boolean
getConsequences(Belief):List<Resource>
getJustifications(Belief):List<Resource>
cacheJustification(Resource):Justification
holds(Justification):boolean
ask(String):boolean
getAntecedents(Justification):List<Resource>
getConsequents(Justification):List<Resource>
getBeliefs():List<Belief>
getLeafBeliefs():List<Belief>
getJustifications():List<Justification>
getSupportListEdges():List<Pair<Justification,Belief>>
getInferredModel():Model
getJustificationEdges():List<Pair<Belief,Justification>>
castToBeliefs(List<Resource>):List<Belief>
castToJustification(List<Resource>):List<Justification>
handleBeliefUpdate(Belief):void
handleJustificationUpdate(Justification):void
getJustificationStatus(Belief):String
getStatus(Justification):String
getStatus(Belief):String
getInList(Justification):List<Belief>
getOutList(Justification):List<Belief>
addSubscriber(ModelSubscriber):void
getBelief(String):Belief
setBeliefState(Belief,boolean):void
getBeliefState(Belief):boolean
getStatements(Belief):List<Statement>
getBeliefStatements():List<Statement>
<<Java Class>>
Tms
knowledgebase
knowledgeBase: Model
NS: String
Graph: String
Tms(Fuseki)
getKnowledgeBase():Model
changed():void
addBelief(String):void
removeBelief(Belief):void
removeJustification(Justification):void
addJustification(String):void
justifies(String,String):void
addToIn(String,String):void
addToIn(String,String,boolean):void
removeFromIn(Belief,Justification):void
addToOut(String,String):void
removeFromOut(Belief,Justification):void
premise(String):void
unpremise(String):void
contradict(String):void
uncontradict(String):void
getStatements(Belief):List<Statement>
getStatements():List<Statement>
getAuxiliaryStatements():List<Statement>
addStatement(Belief,List<String>):void
fetchFromDbpedia(String):void
-db
0..1
<<Java Class>>
Fuseki
knowledgebase
serviceURI: String
service: String
accessor: DatasetAccessor
Fuseki(String,String)
getServiceURI():String
getService():String
-db
0..1
-db
0..1
-tms
0..1
-reasoner
0..1
<<Java Class>>
Main
(default package)
Main()
main(String[]):void
render():void
render(Belief):void
render(Justification):void
renderGraph():void
updateReasoner():void
update():void
handleTmsGraphClick(Map<String,Object>,String):void
handleBeliefTypeChanged(Belief,Resource):void
handleJustificationJustifiesBelief(Justification,String,ViewType):void
handleAddBelief(String):void
handleAddJustification(String):void
handleTmsGraphClickEmpty():void
handleBeliefUpdate(Belief):void
handleJustificationUpdate(Justification):void
handleRemoveInBelief(Belief,Justification,ViewType):void
handleRemoveOutBelief(Belief,Justification,ViewType):void
handleAddInBelief(String,Justification,ViewType):void
handleAddOutBelief(String,Justification,ViewType):void
handleAddDoc(Belief,List<String>,ViewType):void
handleRemoveBelief(Belief,ViewType):void
handleRemoveJustification(Justification,ViewType):void

Figure 2: Class diagram

References

Doyle, J. (1992). Reason maintenance and belief revision: Foundations vs. coherence theories. *Belief revision*, 29:29–51.